Multiple-Valued CCD Circuits

Jon T. Butler, Naval Postgraduate School, United States
Hans G. Kerkhoff, University of Twente, The Netherlands

The foundation of any computing system is the logic on which it is built. Indeed, the impact of computing on modern technology largely results from the success integrated circuit manufacturers have had in implementing complex circuits compactly. During the 1970s, circuit density doubled every year. However, limits in circuit reduction are now being felt, and the pace has slowed to a doubling every two years, with further slowing expected. This has inspired an interest in innovative ways to achieve still smaller circuits.

Currently, important technologies include emitter coupled logic (ECL), n-type metal oxide semiconductor (nMOS), and complementary metal oxide semiconductor. CMOS in particular predominates because of its low power dissipation, high packing density, and reliability. CMOS devices produce very little heat, making the design of compact systems possible. Power is dissipated only when logic values change; the actual storage of a logic value dissipates almost no power.

Unlike CMOS, which uses voltage to store logic values, charge-coupled device (CCD) technology uses charge. Because CCD has many fewer source-drain connections, extremely compact circuits are possible. Also unlike CMOS, multiple values are easily stored, resulting in even more compact circuits. For example, in prototype four-valued CCD designs,[1] logic values are measured by the number of electrons. In such designs, a logic 0 corresponds to zero electrons, a logic 1 to about two million electrons, a logic 2 to about four million electrons, and a logic 3 to about six million electrons. With presently available lithography, a one-micrometer feature size is possible; CCD processing can yield devices where a logic 1 is about 70,000 electrons.[2]

Because CCD devices consume little power, we can realize complex circuits and increase their density with multiple-valued signals.

The prototype designs achieved at the University of Twente in Holland use more than 70,000 electrons because of the university's lack of suitable lithography equipment. Nonetheless, that value is feasible for a logic 1, because noise in modern CCD circuits does not exceed 2,000 electrons.

CCD has been used extensively in the implementation of analog devices such as delay lines and filters, as well as binary devices such as mass memories. Phillips has produced video memories with 318-kilobit and 835-kilobit capabilities.[2,3] Between the two extremes—analog and binary—comes multiple-valued logic.

We have yet to achieve a natural base, 10, for decimally oriented humans. Presently, 6-logic values are achievable[4]; however, there is much interest in 4-logic levels, since 4 is a power of 2 and thus more compatible with binary circuits. The current limitation on the number of logic levels is not noise, but the chip devices that produce logic values. Deviations in the number of electrons due to variations in the processing of CCD chips exceeds the number due to noise and to charge transfer loss.

The major disadvantage of CCD is speed. CCD uses clock pulses to coordinate the arrival of charge, and the interval between clock pulses must be long enough to complete the slowest operation. Thus, the slowest CCD operation determines the total circuit delay. Binary surface-CCD reportedly has achieved sys-

0018-9162/88/0400-0058$01.00 © 1988 IEEE

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
1988		

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Multiple-Valued CCD Circuits	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School,Department of Electrical and Computer Engineering,Monterey,CA,93943**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**12**	

tem clock frequencies of up to 40 megahertz.

In spite of the speed disadvantage, CCD is very useful. Compared to all major technologies, CCD has the lowest power/delay dissipation product for a given logic capability. Therefore, even with a larger delay, the power dissipation is so low that the product of the two parameters is smaller than for any other technology. Moreover, we can mitigate the speed disadvantage by using buried-channel CCD, in which charge flows below the surface. Binary buried-channel CCD reportedly has achieved clock frequencies of 180 megahertz.

Current work in CCD has its origins in the 1940s, when K. Schlesinger demonstrated an analog device consisting of a cascade of capacitors along which charge was passed. Two Dutch researchers, F.L.J. Sangster and K. Teer, implemented this *bucket brigade device* as an integrated circuit in 1969. They used MOS technology for the capacitors, with field-effect transistors serving to shepherd the charge through the device.

Most modern CCDs use a charge transfer mechanism proposed by Boyle and Smith[5] in 1970. Called *multiphase clocking*, it operates in a manner analogous to the pouring of water from one cup to another. The efficiency of this mechanism means that little charge is lost in the transfer. Modern CCD circuits, for example, lose only about one in every 10,000 electrons in each charge transfer.[6]

Another improvement in CCD processing followed shortly in 1972. Up to that point, all CCD circuits had the property that the charge resided on the IC surface. It was observed that most of the time spent in the transfer of charge was in the *last* fraction of each charge packet. L.J.M. Esser proposed that this problem be solved by placing the charge well away from the surface. The new devices, called *peristaltic CCD* or *PCCD*, transfer charge at a level below the surface. The result is a higher speed device with an additional benefit: Because charge is away from the surface, electrons in the charge packet are less affected by surface charge recombination and, thus, fewer electrons are lost. (The loss of enough electrons causes a logic level to deteriorate, making it necessary to include logic level regenerators.)

Most electronic devices rely on voltage or current to encode input and output information. Thus, embedded CCD circuits must convert voltage or current to charge and back again. To do this, a capacitor is used. In a capacitor, a linear relationship exists between the charge stored and the voltage across it, given as $q = Cv$, where C is the capacitance. For example, to convert charge to voltage, a charge is placed on a capacitor inducing a voltage that is then amplified and applied to the circuit output.

Input signals can also be entered into a CCD by the absorption of photons. This property was used in 1971 by G.F. Amelio, W.J. Bertram, Jr., and M.F. Tompsett to realize an image reception CCD. The basic principle is used today in commercial cameras and in low-light-level systems, such as telescopic image reception.

The application of CCD to logic circuits was also recognized early. M.F. Tompsett in 1972 showed how to regenerate binary logic levels in memories and proposed the realization of NAND and NOR gates. The first nonbinary CCD circuit was a four-valued memory developed in 1978 by M. Yamada in Japan. The first four-valued CCD implementations of *logic* operations were developed in 1979. In 1980, Kerkhoff[1] constructed arithmetic and higher level four-valued logic operations, as well as two-valued to four-valued and four-valued to two-valued converters.

In the 10 years since the introduction of multiple-valued CCD, there have been advances in three areas—device technology, logic design, and applications. These three areas define the scope of this tutorial. We begin by presenting the principles on which CCD circuits are based. The use of charge to represent logic values determines the way logic is combined to form higher level functions, and we will show its effect on logic design. Then, we will show specific examples of multiple-valued CCD logic design.

Figure 1 shows the hierarchy on which CCD design is based. At the bottom are basic CCD configurations, specifically configurations that are recognized units in the very-large-scale-integration layout and that have a direct effect on movement of charge in the IC. These configurations include storage wells, clocked transfer gates, and barrier gates.

At the next level are logic operations, of which there are four—addition, constant, multithreshold, and inhibit. Each can be described algebraically, and the four together form a *complete* set; that is, a set of operators from which all logic functions can be formed.

The set of *all* logic functions forms the third level of the hierarchy. The focus here is on efficient realizations of given functions. While problems of logic design in binary have been well studied, it is not yet clear how to design multiple-valued circuits. Consequently, this area attracts intense interest. It is interesting to note that the knowledge that the four configurations at the logic operation level are complete comes from the fact that these configurations are sufficient to realize three functions at the combinational logic function level: (1) sum, (2) minimum, and (3) literal, which form a complete set.

The next two levels represent large-scale circuits. The fourth level corresponds to arithmetic units, such as adders and multipliers, and control units, such as CPU control units. The highest level corresponds to computer systems and processors. This tutorial focuses on the lowest three levels, these being most affected by IC technology. However, we consider several aspects of the higher two levels.

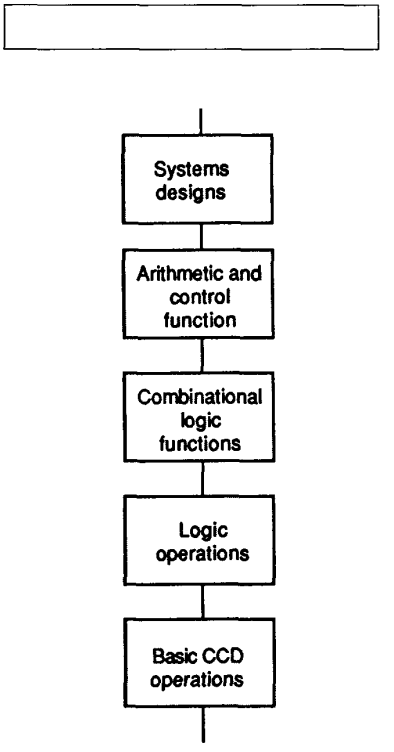


Figure 1. Hierarchy of CCD systems design.

Fundamentals of CCD logic operations

The basic structure for storing a logic value in conventional logic circuits is a path, which in binary is either open or closed. In CCD, logic values are carried as quantities of charge, and the basic structure for storing a logic value is the *potential well*. The well resembles the capacitor used in nMOS dynamic memories. However, in CCD the capacity of the well is an important parameter. Capacity is measured in logic units, which creates an upper bound on the logic value that can be stored. For example, a well with a capacity of two logic units can hold a logic 0, 1, or 2.

The transfer of charge between wells is accomplished by a *clocked transfer gate*. The time of transfer is determined by clock signals that coordinate the movement of charge around the chip. Figure 2 shows how clocked transfer gates perform addition. Potential wells appear as squares and clocked transfer gates as shaded rectangles separating potential wells. In this case, two clocked transfer gates separate potential well 3 from wells 1 and 2. At the clock pulse, the clocked transfer gates move the two charge units in wells 1 and 2 into well 3, where they are added. The process is analogous to pouring the contents of the input wells into the output well. The transfer is unidirectional, with the direction indicated by context.

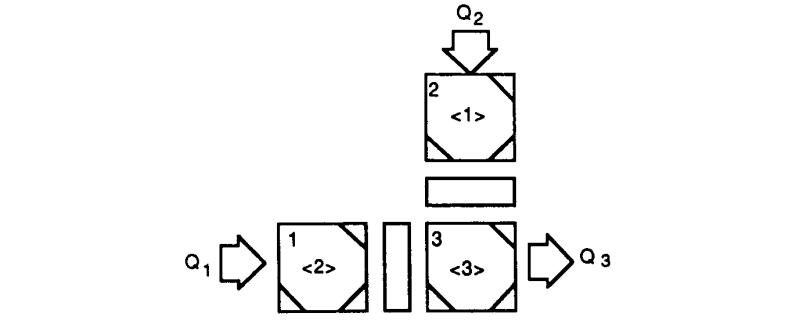


Figure 2. CCD adder.

Basic CCD configurations

Using wells and clocked transfer gates, we can realize higher level functions. Specifically, we discuss four from which it is possible to realize all possible functions. All have been fabricated at the University of Twente, as have all other circuits described in this article.

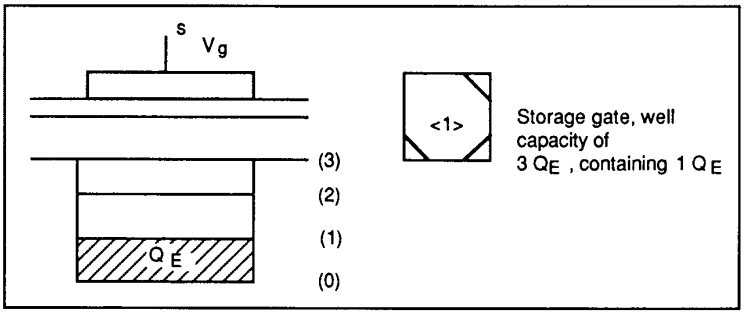
Addition. Because of the fixed size of the well in which the charges are combined, this operation is actually truncated addition. That is, if the sum of the charge put into the sum well exceeds its capacity,

Potential well

The first MVL CCDs were made in a surface CCD process employing double polysilicon interconnections. The IC foundation is a substrate of p-type silicon. Doped silicon-oxide layers are used to establish diffused regions in the substrate and undoped silicon oxide is used for insulation. Two layers of polysilicon form the CCD structures.

The most fundamental CCD structure is the potential well, shown in the accompanying figure. It is a capacitor, one side of which is the substrate and the other an area of polysilicon on top separated by an insulating layer of silicon dioxide.

The well is a region in the substrate where the potential is modified by the voltage applied to the polysilicon. This potential determines the amount of charge the well can contain. The figure shows the side view of a well (left) that can contain a logic 3. A smaller potential would create a well of capacity 2, and a still smaller potential a capacity of 1. The top view (right) shows the potential well as a square. The well's capacity in logic values is indicated by the number of notches in corners. For example, the three notches indicate a capacity of logic 3.



Potential well.

the excess charge is lost. Thus, the output is the sum of the inputs except where that sum exceeds the sum well capacity c, in which case the output is c.

Because of its relative simplicity, addition is a widely used operation. It is a key operation in the logic design algorithms discussed later.

Multithreshold. The multithreshold circuit acts like a threshold gate producing an output only when the input exceeds some predetermined value. Figure 3 shows an example of such a circuit. Here there is a single input and three outputs. At the clock pulse, the input charge (to the left of the clocked transfer gate, represented by the vertical shaded bar) flows to wells 1, 3, and 5. Separating these wells are fixed barrier gates represented by nonfilled rectangles.

Like clocked barrier gates, fixed barrier gates facilitate the transfer of charge. Unlike clocked barrier gates, the transfer can occur any time, not just at clock pulses.

The operation of a multithreshold cir-

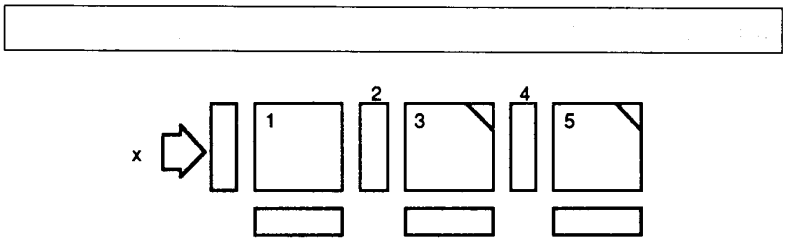


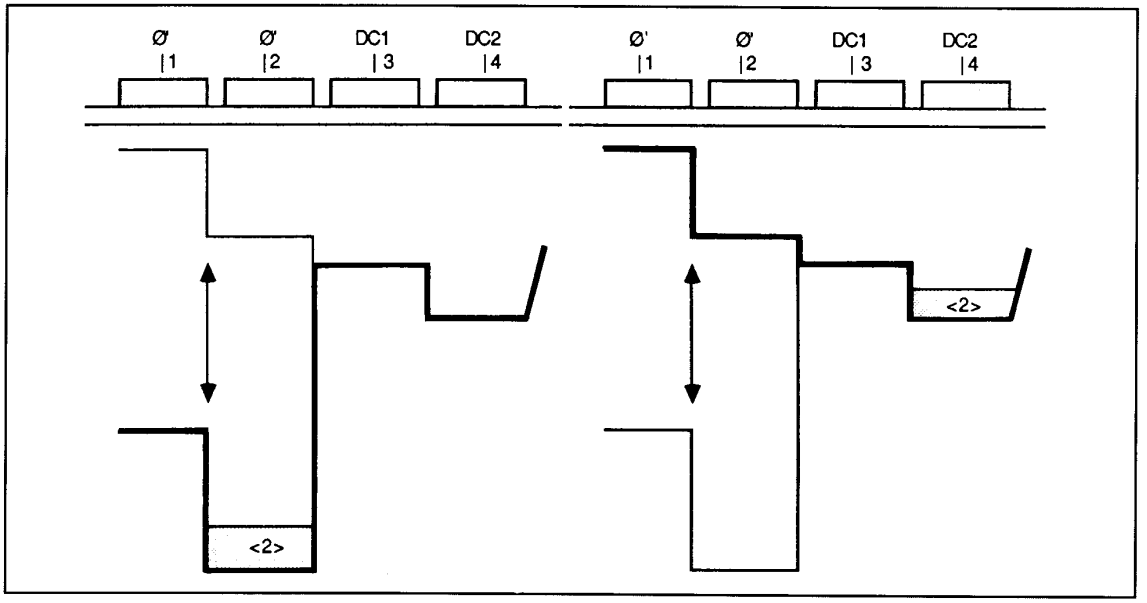
Figure 3. A multithreshold circuit.

Clocked transfer gate

The transfer of charge between two wells is performed in CCD by a gate that drains the first well of its charge, placing it into the second well. A system clock signal activates the transfer.

The top view of a clocked transfer gate is a rectangle, shaded to indicate that it is driven by the system clock. A side view of the gate, as seen in the accompanying figure, shows the voltage values during transfer. The top part of this view shows the two wells and the transfer gate as a set of four capacitors. Two capacitors, 1 and 2, receive a voltage value, ϕ' and ϕ, respectively, that varies between the two values shown below it. The other two capacitors, 3 and 4, receive a constant voltage value.

Prior to the clock pulse (part (a) of the figure), charge resides under capacitor 2, having come in from the left. The clock pulse raises the voltage under capacitors 1 and 2, causing the charge to flow to 4. Part (b) shows the result. The voltage on capacitor 1 is always higher than that of 2, preventing charge from flowing toward the left. For the same reason, the voltage on capacitor 3 is higher than on 4.



Charge transfer in the clocked transfer gate, showing (a) charge before clock pulse, and (b) charge after clock pulse.

cuit resembles the filling of an ice cube tray. That is, charge flows first to well 1. If this is filled to capacity, excess charge flows to well 3. If this fills to capacity, the remaining excess flows to well 5. Thus, wells 1, 3, and 5 produce a logic 1 output if the input logic value is equal to or greater than 1, 2, and 3, respectively. In this way, the multithreshold circuit acts like a threshold element, producing an output depending on whether a well capacity is exceeded or not.

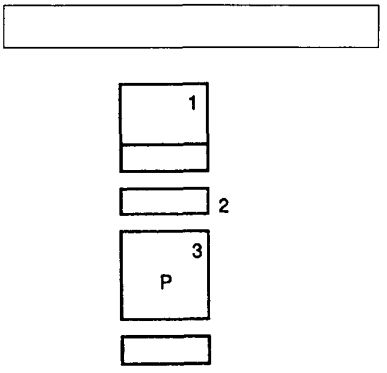


Figure 4. Constant generator.

Constant generator. Figure 4 shows how a constant logic value is produced. Well 1 is a source of charge produced at a clock pulse, as indicated by the shaded bar embedded in well 1. The fixed barrier gate 2, in effect, determines the capacity of well 3, shown here as p. At the clock pulse, well 3 is filled to capacity by the charge source 1. At the next clock pulse, this is transferred out of the constant generator.

Inhibit gate. An important part of the inhibit gate is the floating sense gate. The floating sense gate provides *action-at-a-distance* capability in CCD. It consists of a special potential well called a *sense well* and an attached barrier gate. A wire at the metalization layer connects the sense well and the barrier, so the barrier can be placed away from the sense well.

When the sense well contains a nonzero logic value, the attached barrier gate is high. Conversely, an empty sense well produces a lowered barrier. Figure 5 shows the use of the floating sense gate in an inhibit gate.

The inhibit gate operates as a switch for an input charge. This charge flows out one of two outputs depending on the value of another input, the controlling input. As shown in Figure 5, the main input charge is applied at the top. Barrier gate 3 is high if the controlling input charge in gate 7 is nonzero, so the main input charge in well 2 is inhibited from moving into well 4 (thus the name, inhibit gate); at the clock pulse, it moves right and out of the circuit. On the other hand, if gate 7 has no charge, then barrier gate 3 is down and charge from the top well flows through well 2 across barrier gate 3 and into well 4, where it is clocked out at the lower output.

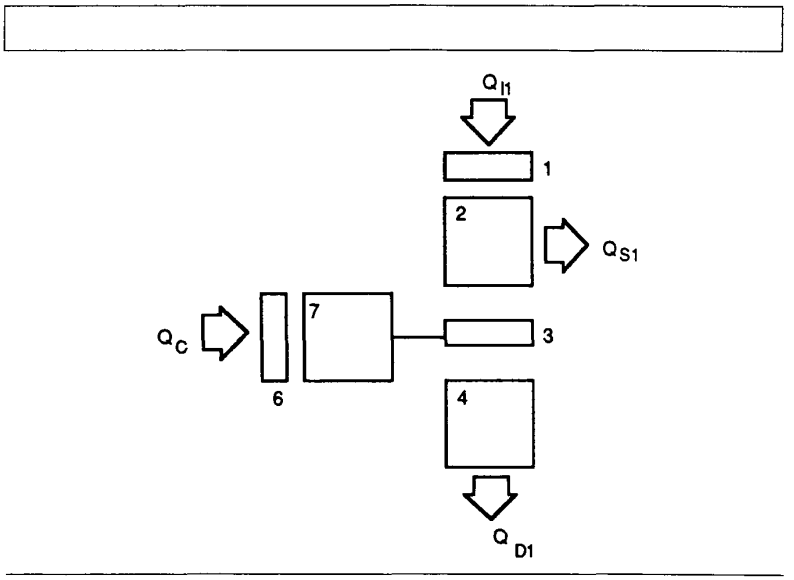


Figure 5. Inhibit gate.

Programmable logic arrays

The capacity of a well is easily changed, since it depends on the applied voltage. As a result, we can "program" the logic function of a CCD circuit by an appropriate choice of voltages. This is the basis of the CCD programmable logic array, or CCD-PLA.[4]

Figure 6 shows the basic CCD-PLA circuit. Here, input x is applied to two multithreshold circuits. In the uppermost multithreshold circuit are two wells, one of capacity a and one of capacity b. At the clock pulse, the charge from x flows right into well a. If $x > a$, then the excess, $x-a$, flows into the adjacent well. At the next clock pulse, $x-a$ in this well flows onto the adder, shown as a line on the right in Figure 6. The diagram to the right of the upper multithreshold circuit shows how the output $x-a$ applied to the adder depends on the input x and a.

The lower multithreshold circuit behaves in the same way except for the complement circuit represented by a triangle. This circuit forms the identical function of the complement circuit shown in Figure 7. However, its implementation in Figure 6 is much simpler. Since x is an external signal, it is encoded as a voltage. It is in the voltage-to-charge conversion that the complement is realized. In fact, the complement is realized with exactly the same circuitry as a noncomplementing voltage-to-charge converter.[1] The diagram to the right of the lower multithreshold circuit shows that its output function is the reverse of that of the upper multithreshold circuit.

In Figure 6, the output is derived from the column that realizes a product term. Figure 7 shows the circuit that the column drives. That is, the column is applied to a sensitive sense amplifier that, in turn, controls the barrier of an inhibit gate consisting of a source well and a well f with programmable capacity f. Thus, when there is charge on the column, the barrier is high, blocking charge from the source

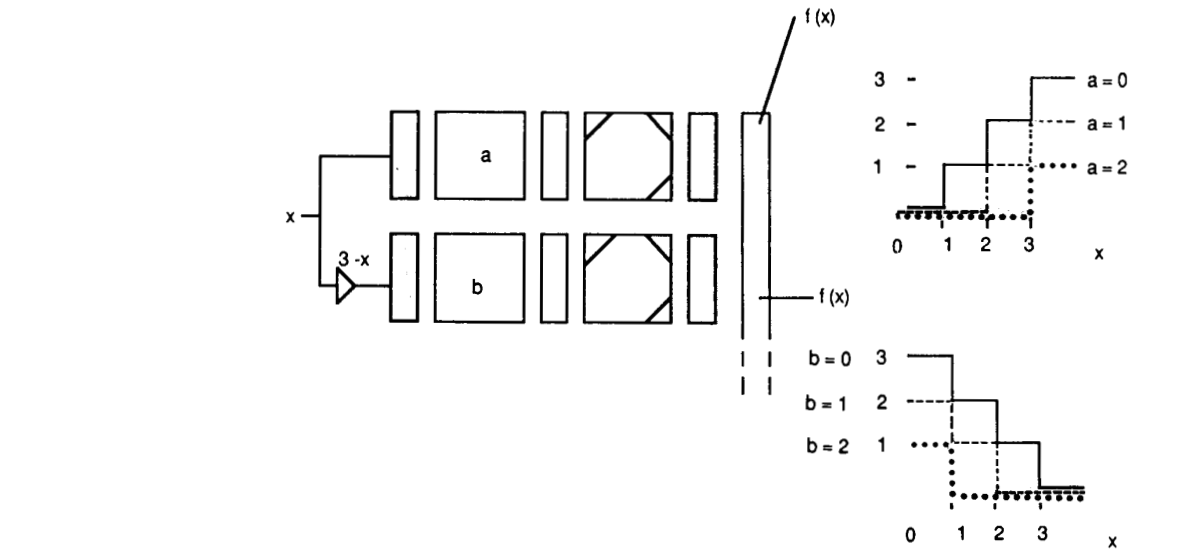


Figure 6. Basic CCD-PLA circuit.

well to well f. When there is no charge, the flow is not blocked and well f fills to its capacity f. At the next clock pulse, a quantity of charge, f, is transferred to the output column, where it is summed with the outputs of all other columns.

The output of the PLA column is f only if the outputs to all basic CCD-PLA circuits of that column are 0. A basic CCD-PLA produces a 0 when both multithreshold circuits produce a 0. Because of the staircase functions produced by the multithreshold circuits, a 0 is produced only for x in the range between the lowest steps of the two staircase waveforms. For example, if, in Figure 6, $a = 2$ and $b = 2$, then the basic CCD-PLA circuit corresponding to x produces a logic 0 when $1 \leq x \leq 2$ and a logic 1 otherwise.

This logic operation is similar to the following. Let x_i be a four-valued variable. Thus, $0 \leq x_i \leq 3$. The *literal function* is defined as

$$f(x_i) = {}^{a}x_i^{b} = \begin{cases} r-1 \text{ if } a \leq x_i \leq b \\ 0 \text{ otherwise} \end{cases}$$

Since the output values of the literal operation are just 0 and $r-1$, this operation yields binary output values given the full range of multivalued input values. Specifically, the output assumes the higher value only when the input values occur in the range $a \leq x_i \leq b$. It is also called a *decisive* function because it executes a decision operation, placing logic values into two categories, those which produce 0 and those which produce $r-1$.

The *Min operation* is defined as

$$x \cdot y = Min(x, y)$$

The term *min* describes the fact that the output value is the minimum of the two input values. The output of the inhibit controlled by the sum of the basic CCD-PLA circuits will be f if there is no charge in the sum and 0 if there is a nonzero charge. Thus, the CCD-PLA column produces a logic function that can be written as

$$f \cdot {}^{a_1}x_1^{b_1} \cdot {}^{a_2}x_2^{b_2} \cdot \ \ldots \ \cdot {}^{a_n}x_n^{b_n}$$

This is called a *product term* because of its resemblance to multiplication. The product term produces logic value f if $a_i \leq x_i \leq b_i$ for all i, and produces 0 otherwise.

The output column produces the (truncated) sum of the column outputs, and the resulting sum-of-products is the PLA output. We know that *any* function has at least one sum-of-products expression. Thus, any function can be realized by a CCD-PLA of the appropriate size.

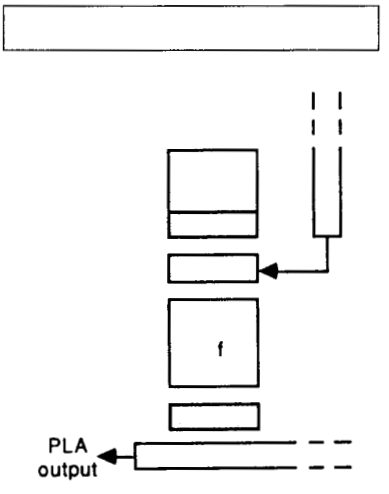


Figure 7. CCD-PLA column output circuit.

Figure 8 shows a photomicrograph of the basic CCD-PLA circuit. The object left-of-center, which is the junction for many leads, is the basic CCD-PLA cell. The two multithreshold circuits occur in

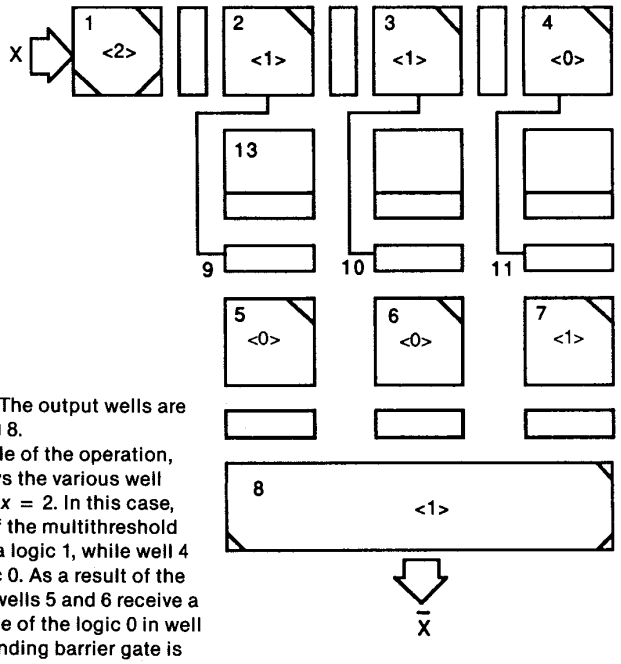
Complement circuit

A binary complement circuit is a single-input, single-output device that realizes the expression $q = (r-1)-x$, where x is the input, q is the output, and r is the radix. When $r = 2$, this corresponds to the binary complement operation $q = \bar{x}$.

The accompanying figure shows a CCD realization of a four-valued complementer ($r = 4$). The circuit consists of two parts, a multithreshold circuit and three inhibit circuits that drive an adder that provides the circuit's output. The multithreshold circuit serves to divide four-valued variable x into three two-valued quantities. Specifically, if x is a logic 3, then a logic 1 exists in each of the wells 2, 3, and 4. If x is a logic 2, then a logic 1 exists in wells 2 and 3 only. If x is a logic 1, then a logic 1 exists only in well 2.

Each of the wells 2, 3, and 4 is a sense well for a barrier gate separating a source of charge from a well of capacity one. Each of these three circuits forms a binary complement, with the barrier passing a logic 1 if the corresponding sense well has a 0 and vice versa. The output wells are summed in well 8.

As an example of the operation, the figure shows the various well contents when $x = 2$. In this case, wells 2 and 3 of the multithreshold circuit receive a logic 1, while well 4 receives a logic 0. As a result of the inhibit action, wells 5 and 6 receive a logic 0. Because of the logic 0 in well 4, the corresponding barrier gate is down and well 7 receives a logic 1. Thus, the output, which is the sum of the wells, is 1.



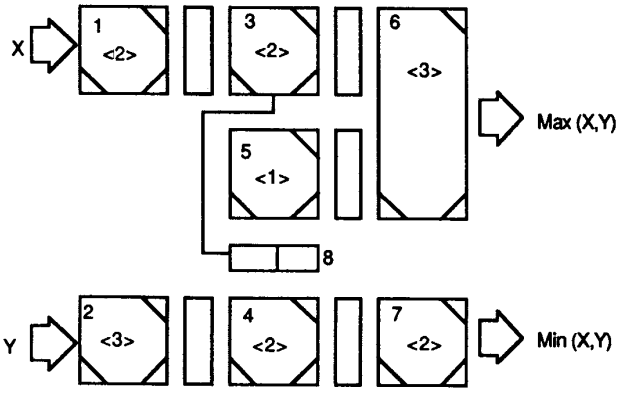
Four-valued complement circuit.

Minimum/maximum circuit

Shown in the accompanying figure is a CCD circuit that realizes the maximum and minimum of two logic values x and y. The circuit, in effect, acts as a switch, placing at $Max(x, y)$ the larger of the two values and at $Min(x, y)$ the smaller. It employs a unique capability of CCD.

The two inputs x and y appear in wells 1 and 2 initially. 1 and 2 are transferred through the clocked transfer gates shown into wells 3 and 4, respectively. Well 3 is a sense well for a special barrier well 8. That is, the barrier height of 8 is directly proportional to the amount of charge in well 3. Furthermore, this barrier determines the capacity of well 4. Specifically, if well 3 contains logic value x, then the capacity of well 4 is x. Any charge from y in excess of x flows over the barrier into well 5. In this example, $x = 2$ and $y = 3$. Thus, the capacity of well 4 is two. Since y exceeds this by one, a logic 1 flows over the barrier into well 5. At the clock pulse, the sum of the contents of wells 3 and 5 appears in well 6 while the contents of well 4 are transferred into well 7.

Well 7, $Min(x, y)$, contains the minimum of the two values since if $x > y$, the value of y is transferred undiminished into well 7. Conversely, well 6, $Max(x, y)$, receives x if $x > y$ and receives $x + y - x = y$ if $y > x$.



The maximum/minimum circuit.

the top and bottom halves of this part of the figure. The large uniform square near the upper right-hand corner of the picture is a capacitor used to measure characteristics of the substrate. To its left and slightly lower is a transistor used to measure threshold voltages of the chip. In the lower right-hand corner is the balanced sense amplifier used in detecting charge on the column output.

It is interesting that the CCD-PLA structure remains unchanged if we choose a larger or smaller radix. The only change needed is in the number of voltages for programming well capacities. In such a circuit, we choose the highest radix needed to provide sufficient separation between logic levels, since this yields the highest logical complexity per unit of chip area. That is, increasing the radix is effectively the same as making the circuit more compact.

Logic design of MVL-CCD

We can combine basic CCD gate configurations to form higher level operations. While basic CCD configurations model closely the physical behavior of the CCD circuit, it is easier to design using the higher level operations. In fact, our knowledge that the four basic CCD operations are sufficient to realize all functions comes from the fact that such operations realize a set of higher level operations that are known to realize all functions. Specifically, the four basic operations can be used to realize the Max, Min, and literal operations, a set we know will realize any logic function. We show two such synthesis techniques for multiple-valued CCD circuits; other techniques are discussed in Kerkhoff.[1] Unlike binary, which has a long history of logic algorithm research, we know considerably less about multiple-valued logic design.

Having defined the four basic CCD configurations and their operation, we are in a position to consider the problem of realizing logic circuits from these configurations. The problem is nontrivial because of the extreme complexity of computing systems. It follows that a design approach must be easily implemented by a computer program and must produce reasonably efficient designs.

One approach introduced by Kerkhoff[1] is called the *cost table* technique because each entry has a cost associated with it. To realize a given function, a selection is made from the table so that the selected functions combine to produce the given function at the lowest possible cost.

The cost table approach combines the advantage of custom design (efficient realizations) with a formal procedure that can be implemented by a computer program. There is also a flexibility associated with the choice of cost. While the presently used costs correlate closely with the IC chip area, other possible costs are (1) delay and (2) yield. For example, in the case of the latter cost, some CCD operations cannot be produced as reliably as others, so a smaller percentage of manufactured circuits will operate within specifications.

Because the realization of a given function can be accomplished by an appropriate choice of functions from the cost table, one synthesis technique is exhaustive enumeration. That is, we simply enumerate all possible ways to realize a given function and then choose the one with the least cost.

While practical for one-input, one-output functions, exhaustive enumeration is prohibitively time-consuming for functions of two or more variables. This observation has inspired research into other approaches, such as directed search, which uses an approximation to the costs used in previous work. The approximation has the advantage that costs are extracted directly from the function, allowing a formal process in which certain realizations of the given function can be quickly eliminated as nonminimum realizations.

The logic design of binary PLAs has received considerable attention. The problem of finding a minimal solution is known to be NP-complete, which has inspired research on heuristic approaches to binary PLA minimization.

A similar emphasis on heuristic approaches to multiple-valued PLAs exists. At present, there are three similar approaches. All are two-step processes. The first step is to select a minterm—an assignment of values to the variables such that the given function is nonzero. For such an assignment, at least one column produces a nonzero value. The second step is to find a product term that covers the selected minterm. This corresponds to choosing the function realized by one column that produces a nonzero value when the selected assignment of input values is applied to the PLA. The product term is subtracted from the function and the process is repeated on the resulting function.

In the first method, used by Pomper and Armstrong,[7] the minterm is selected randomly and the product term is selected as the largest that covers it.

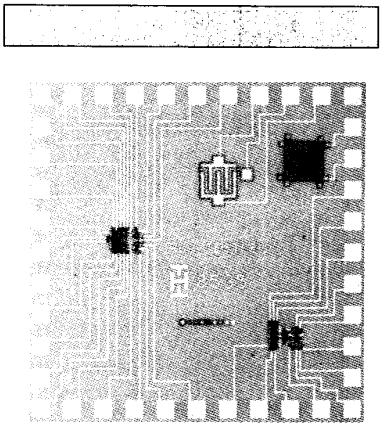


Figure 8. Photomicrograph of the basic CCD-PLA circuit.

In the second method, used by Besslich,[8] the minterm is selected as the most *isolated* minterm—the minterm farthest away from all other minterms. The reduction of the most isolated minterm is an attempt to find an essential product term, a term that appears in all minimal sum-of-product expressions. The product term chosen is one that tends to drive the largest number of minterms to 0 when the selected product term is subtracted.

The third method, used by Dueck and Miller,[9] selects the minterm in a similar way to that of Besslich.[8] However, the product term selected is one that tends to produce the most equal balance among nearby minterms; that is, it tends to choose product terms that when subtracted tend to produce equal logic values in nearby minterms.

All three techniques apply or can be easily modified to apply to the minimization of sum-of-products expressions, where "sum" refers to the truncated sum of the CCD sum operation. Interestingly, the truncated sum operation has been shown to be superior to the Max operation over two classes of multiple-valued logic functions.[10] That is, on the average, fewer product terms are required for the realization of functions when the sum is used compared to when the Max is used. We know also that there are functions for which the Max operation would require significantly fewer products than the sum.

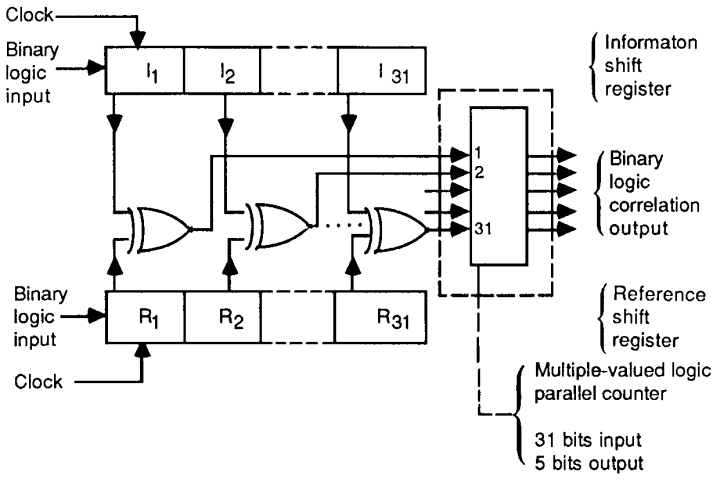


Figure 9. Circuit for a parallel correlator circuit.

MVL CCD-PLA design

In a binary PLA, a given function is realized as the OR of product terms, where a product term is the AND of binary variables or their complement. For example, the two-variable exclusive OR function is realized as the OR of two product terms, $x_1\bar{x}_2 + \bar{x}_1x_2$, where + is OR. In a multiple-valued PLA, the OR function is replaced by the truncated sum (also represented as +), which is normal addition except when the result exceeds $r-1$, the highest logic value, in which case the result is taken as $r-1$.

A product term

$$c \cdot {}^{a_1}x_1^{b_1} \cdot {}^{a_2}x_2^{b_2} \cdot \ \ldots \ \cdot {}^{a_n}x_n^{b_n}$$

in multiple-valued logic is more complicated because of the larger number of logic values. The additional values affect both the product term value and the form of the one-variable functions. The product term value is specified by a constant c, where c is one of the nonzero logic values. The one-variable literal function ${}^{a_i}x_i^{b_i}$ provides a discrimination among values of x_i. When x_i is between a_i and b_i, the function is $r-1$, the highest nonzero logic value. Outside this range, the function is 0. When x_i is a binary-valued variable, there are only two nontrivial literal functions, ${}^1x^1 = x$ and ${}^0x^0 = \bar{x}$.

Each column in a CCD-PLA realizes a product term

$$c \cdot {}^{a_1}x_1^{b_1} \cdot {}^{a_2}x_2^{b_2} \cdot \ \ldots \ \cdot {}^{a_n}x_n^{b_n}$$

through a mechanism that depends on the presence or absence of charge deposited on the column. That is, each basic CCD-PLA circuit deposits a nonzero charge on the column if and only if x_i is outside the range a_i through b_i. Thus, a column has no charge if and only if *all* variables are within their specified range.

When no charge is on the column, the inhibit input, which is driven by the column, produces at its output a charge corresponding to a logic c. This is precisely the function

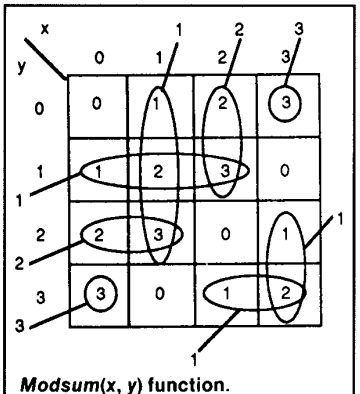
$$c \cdot {}^{a_1}x_1^{b_1} \cdot {}^{a_2}x_2^{b_2} \cdot \ \ldots \ \cdot {}^{a_n}x_n^{b_n}$$

The column output values are all summed to form the PLA output.

The accompanying figure shows how the two-variable function *Modsum*(x, y) is realized as a sum-of-products function of a CCD-PLA. This function is analogous to the exclusive OR function, which is *Modsum*(x, y) in binary logic. The circles in the figure represent product terms whose constant value is shown nearby. For example, the circle surrounding the logic values in the squares $x = 2$ and $0 \le y \le 1$ corresponds to the product term $2 \cdot {}^2x^2 \cdot {}^0y^1$. The sum-of-products expression for this realization is

$$\textit{Modsum}(x, y) = 1 \cdot {}^1x^1 \cdot {}^0y^2 + 1 \cdot {}^0x^2 \cdot {}^1y^1 + 2 \cdot {}^2x^2 \cdot {}^0y^1 + 2 \cdot {}^0x^1 \cdot {}^2y^2 + 3 \cdot {}^0x^0 \cdot {}^3y^3 + 3 \cdot {}^3x^3 \cdot {}^0y^0 + 1 \cdot {}^2x^3 \cdot {}^3y^3 + 1 \cdot {}^3x^{3 \cdot 2}y^3$$

There are six product terms, so a six-column CCD-PLA is needed.



***Modsum*(x, y) function.**

But then, we know that the converse is also true.

These techniques are a necessary part of CAD tools for the design of multiple-valued PLAs. A CAD tool for the design of CCD-PLAs already exists,[11] the first known tool for multiple-valued logic design.

Applications

As one example of a CCD logic design application, consider a parallel correlator. Such circuits are useful in sonar, radar, and communications systems where an incoming signal is compared with a known signal. If the two are highly correlated, then the incoming signal is judged to be an instance of the known signal. This process allows us to detect weak signals buried in noise.

Figure 9 shows a correlator for a 31-bit binary signal that could, for example, be a discretized version of a radar return signal. The binary circuit input is shifted into the information shift register, while the reference signal is shifted into the reference shift register. The two streams are compared bit-by-bit by 31 exclusive-OR gates. A logic 1 at the output indicates that the two input bits are identical. By counting the number of bits at the output of these gates, a measure of the correlation is obtained. This is done by a parallel counter with 31 inputs and five outputs, where the output is a binary number representing the number of 1's on the inputs. A compact realization of the parallel counter is achieved through the use of four-valued CCD circuits. Figure 10 shows that the counter consists of four 7-bit adders (7BA) and three quaternary full adders (QFA).

Each seven-bit and quaternary full adder produces two outputs, a four-valued output representing the sum and a binary output representing the carry. The input to the quaternary full adder is a pair of four-valued variables and a binary carry input. Each seven-bit binary adder has seven binary inputs and is a quaternary full adder in which the two four-valued inputs are replaced by six binary inputs.

The circuits for the binary and quaternary full adders in CCD have the same complexity, so the circuit shown in Figure 10 uses approximately half the circuitry required by an all-binary design. Also, there is a considerable reduction in interconnections. Since most of the area of typical binary VLSI circuits is devoted to interconnections, this is significant.

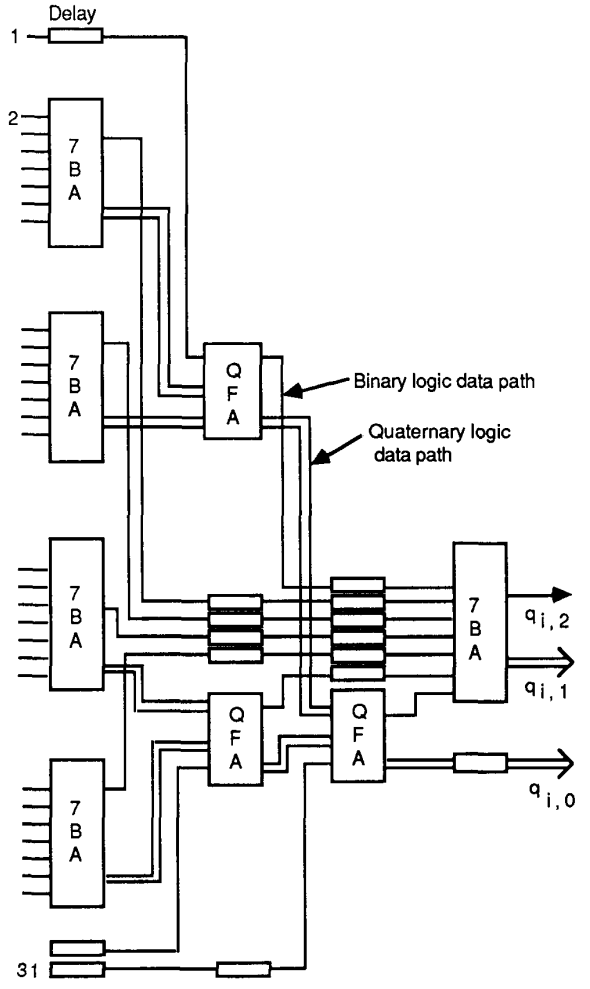


Figure 10. A 31-bit parallel counter using seven-bit adders and quaternary adders.

Another application that CCD suits admirably is in image processing. The use of multiple-valued logic is especially attractive because it eliminates the decoding operation between a gray-level image and the binary representation of that gray level.

Kerkhoff, Zijlstra, and Onneweer[12] showed an architecture for a multivalued image processor using pipelined CMOS and CCD. Unlike conventional designs where image reception and storage are separate, this MVL processor uses the light-reception CCD to store the image. The heart of the processor is a multivalued pixel analyzer that determines subsequent pixel values as a function of pixels in a 3×3 window surrounding the pixel. For example, it is possible to shrink objects to a simple cell, allowing an automated counting technique. Also, alphanumeric characters can be skeletonized to stick-like figures applicable to pattern-recognition programs.

Because little power is consumed in CCD, complex circuits can be realized with very small interdevice distances. The use of multiple-valued signals further improves logic density. While six-valued CCD is possible, four-valued CCD evokes the most interest because of its compatibility with binary circuits.

The realization of four-valued CCD logic circuits is accomplished through four basic CCD configurations. These form a complete set from which any function can be realized. One promising design approach is the cost table technique. Here, a given function is realized by selecting functions from the table and combining them. Another approach is the PLA. This fixed structure CCD circuit is programmed to realize a given function by adjusting well capacities.

The major disadvantages of CCD are speed and the necessity of clock signals. Thus, CCD has application where speed is not as critical as complexity. CCD will continue to be used in image processing, an area where it is the dominant technology. Also, CCD will continue to find application in extremely compact memory circuits. Past successes include 318-kilobit and 856-kilobit video memories by Phillips.[2,3] It is expected that, as logic is merged with memory, there will be increased use of CCD in the implementation of combinational logic.

The cost of CCD processing, which was high 10 years ago compared to the cost of CMOS, is now comparable. There is considerable interest in merging the two technologies, with the prospect of finding a hybrid that takes advantage of the best of both technologies. There is also interest in using CCD in implementing neural networks, where high-speed computation is not so critical as in conventional computing. For example, a serial CCD neural net has been fabricated at the University of Southern California.

The full potential of multiple-valued CCD will require an understanding of the design process. It is not yet clear what basic configurations and what design techniques provide the best trade-off between flexibility and efficiency of realization. The question has been answered for the most part in binary, because of the small number of such operations and because of a long period of research. However, even in three-valued logic there are no definitive answers for the present. □

Acknowledgments

We wish to thank the North Atlantic Treaty Organization, the Dutch Foundation for Fundamental Research on Matter (FOM), and the National Science Foundation for grants that supported research reported in this article.

References

1. H.G. Kerkhoff, *Theory, Design, and Applications of Digital Charge-Coupled Devices*, PhD Thesis, Twente University of Technology, Enschede, The Netherlands, Apr. 1984.
2. M.J.M. Pelgrom et al. "Digital Field Memory for Television Receivers," *IEEE Trans. Consumer Electronics*, Vol. CE-29, Aug.

1983, pp. 242-250, and personal communications with M.J.M. Pelgrom.
3. H. Veendrick et al., "835Kb CCD Video Serial Memory," *Proc. Int'l Solid State Circuits Conf.*, Feb. 1988.
4. H.G. Kerkhoff and J.T. Butler, "Design of a High-Radix Programmable Logic Array Using Profiled Peristaltic Charge-Coupled Devices," *Proc. 16th Int'l Symp. Multiple-Valued Logic*, May 1986, pp. 128-136.
5. W.S. Boyle and G.E. Smith, "Charge-Coupled Semiconductor Devices," *Bell System Technical J.*, Vol. 49, Apr. 1970, pp. 587-593.
6. H.S. Abdel-Aty Zohdy, S.G. Chamberlain, and L.A.K. Watt, "Limitations of Multilevel Storage in Charge-Coupled Devices," *IEEE Trans. Electronic Devices*, Vol. ED-27, Sept. 1980, pp. 1733-1743.
7. G. Pomper and J.R. Armstrong, "Representation of Multivalued Functions Using the Direct Cover Method," *IEEE Trans. Computers*, Vol. C-30, Sept. 1981, pp. 674-679.
8. P.W. Besslich, "Heuristic Minimization of MVL Functions: A Direct Cover Approach," *IEEE Trans. Computers*, Vol. C-35, Feb. 1986, pp. 134-144.
9. G.W. Dueck and D.M. Miller, "A Direct Cover MVL Minimization Using the Truncated Sum," *Proc. 17th Int'l Symp. Multiple-Valued Logic*, May 1987, pp. 221-227.
10. E.A. Bender, J.T. Butler, and H.G. Kerkhoff, "Comparing the Sum with the Max for use in Four-Valued PLAs," *Proc. 15th Int'l Symp. Multiple-Valued Logic*, May 1985, pp. 30-35.
11. H.G. Kerkhoff and J.T. Butler, "A Module Compiler for High-Radix CCD-PLAs," preprint.
12. H.G. Kerkhoff, P. Zijlstra, and S.P. Onneweer, "A Pipelined MVL Image-Processing System Based on a Combined CCD-CMOS Process," *Proc. 15th Int'l Symp. Multiple-Valued Logic*, May 1985, pp. 361-368.

Jon T. Butler is guest editor of this issue of *Computer*. His biography and photo appear at the end of the Guest Editor's Introduction.

Hans G. Kerkhoff has been a staff member of the IC-Technology and Electronics Group of the University of Twente, The Netherlands, since 1977. He is currently interested in testable computer-aided design of CMOS VLSI dedicated digital signal processors.

Kerkhoff received the MSc degree in electronic engineering from the Technical University of Delft, The Netherlands, in 1977. He obtained the PhD degree in microelectronics design from the University of Twente in 1984.

Readers interested in receiving reprints of papers on multiple-valued CCD should circle Reader Service Number 160 on the reader service card.

Readers may write to Butler (Code 62-Bu) at the Dept. of Electrical and Computer Engineering, Naval Postgraduate School, Monterey, CA 93943-5100.